

# Trabajo Práctico: Lanzando el Caber

## Clase Lanzamiento

La clase lanzamiento representa un tiro de caber realizado por un lanzador de la competencia. El lanzamiento está compuesto por una distancia, expresada en metros, que se mide en línea recta a la carrera del lanzador y un ángulo de aterrizaje, expresado en grados, del caber.

Esta clase posee las siguientes responsabilidades:

- **esValido**: Indica si el lanzamiento es válido. Se considera un lanzamiento válido a aquel que tenga un ángulo de aterrizaje entre las 9 y 3 horas.
- **getDistancia**: get para poder obtener el atributo distancia de la instancia.
- **getAngulo**: get para poder obtener el atributo ángulo de la instancia.

## Clase Lanzador

La clase lanzador representa a un participante de la competencia de lanzamiento de caber, dicho participante cuenta con tres lanzamientos.

Esta clase tiene como responsabilidades:

- **obtenerVarianzaD**: Obtiene el promedio de la distancia de los tres lanzamientos del competidor, y luego la varianza.
- **obtenerVarianzaA**: Obtiene el promedio de los ángulos de los tres lanzamientos del competidor, y luego la varianza.
- **distanciaTotal**: Se utiliza para obtener la distancia total de los tres lanzamientos de cada competidor. Si el lanzamiento fue inválido, se considera que la distancia fue cero. Si el ángulo del cáber se encuentra entre las 9 y las 11, o entre la 1 y las 3, se considerará el 80% de dicha distancia. Si el ángulo se encuentra entre las 11 y la 1, se considerará el largo total del lanzamiento.
- **getModuloConsistencia**: retorna el módulo de la suma ambas varianzas (ángulo y distancia)
- **isTodosValidos**: retorna el valor del atributo isTodosValidos
- **getDistanciaLanzamientos**: retorna el valor del atributo distanciaLanzamientos.
- **getVarD**: retorna el atributo getVarD
- **getVarA**: retorna el atributo getVarA

## Clase Competencia

Esta clase es una representación de la competencia de lanzamiento de cabers, en la cual pueden participar entre tres y un millón de lanzadores. La competencia presenta dos modalidades en las cuales todos sus participantes competirán, competencia por consistencia y por distancia, cada una de ellas tendrá un podio de tres o menos participantes.

La competencia tiene la responsabilidad de:

- **obtenerPodioCons**: Arma el podio con los 3 lanzadores ganadores. El mismo se determina de acuerdo a la consistencia de los tiros (sostenidamente lance el caber distancias similares, con ángulos consistentes). No se considera

a los lanzadores que tengan algún tiro inválido.

- **obtenerPodioDist:** Arma el podio con los 3 lanzadores ganadores. El mismo se determina de acuerdo a la suma de la distancia total entre los tres lanzamientos.
- **getPodioDistancia:** retorna el podio por distancia de lanzamientos.
- **getPodioConsistencia:** retorna el podio por consistencia de la lanzamientos.

## Clase Archivo

Esta clase se utiliza para leer el archivo de entrada y escribir el de salida con los podios

La clase Archivo tiene la responsabilidad de:

- **leerYCargarDatosDe:** Lee el archivo de entrada con los competidores y sus lanzamientos. Estos datos se irán cargando en un vector de tipo Lanzador.
- **escribirResultadosEn:** Escribe en un archivo de salida los podios por consistencia y distancia. Este método recibe como parámetro el nombre que se le quiere dar al archivo de salida, y ambos podios.